P1 1075029

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

October 06, 2003

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: ***60/409,425***
FILING DATE: ***September 10, 2002***
RELATED PCT APPLICATION NUMBER: *PCT/US03/28149*

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

M. Tarver

M. TARVER
Certifying Officer

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

DATE OF DEPOSIT: 9/10/02

9-11-02 EL803834685US 60/409425 .09/10/02

Please type a plus sign (+) inside this box → [+]

PTO/SB/16 (02-01)
Approved for use through10/31/2002 OMB 0651-0032
U S Patent and Trademark Office, U S DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No. EL 803834685US

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Joseph Samuel	Blevins, Sr.	Export, Pennsylvania

☐ *Additional inventors are being named on the ___ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

HOT STANDBY METHOD AND APPARATUS

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☑ Customer Number 003705 → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☑ Firm or Individual Name	Kirk D. Houser				
Address	Eckert Seamans Cherin & Mellott, LLC				
Address	600 Grant Street, 44th Floor				
City	Pittsburgh	State	PA	ZIP	15219
Country	US	Telephone	412.566.6083	Fax	412.566.6099

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification *Number of Pages* 25 ☐ CD(s), Number

☑ Drawing(s) *Number of Sheets* 1 ☐ Other (specify)

☐ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees

☑ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 02-2556

☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($): $160.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☑ No

☐ Yes, the name of the U S Government agency and the Government contract number are _____

Respectfully submitted,

SIGNATURE _____ Date 9 / 10 / 02

TYPED or PRINTED NAME Kirk D. Houser REGISTRATION NO. *(if appropriate)* 37,357

TELEPHONE 412.566.6083 Docket Number: 283359-00360

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

HOT STANDBY METHOD AND APPARATUS

1.0 INTRODUCTION

Many of the past hindrances to develop a hot standby Microlok have been due to an inability to remain focused on the fundamental reason for a hot standby. A hot standby is for the purpose of having HARDWARE backup not LOGIC backup. Since one Microlok unit is capable of providing failsafe operation, an additional unit is not for the purpose of making the system more failsafe, it is simply providing a backup system that can be utilized until a maintainer can be dispatched to repair the hardware of the primary unit.

The hot standby method and apparatus disclosed herein is organized in such a way that it can easily be incorporated into any Microlok application program to produce a hot standby.

The method and apparatus disclosed herein is applied to an Interlocking Control System (ICS), such as the Microlok® railroad interlocking control system for railroad switching and signaling, as described in U.S. Patent No. 5,301,906, which is incorporated herein by reference. Although Microlok units are disclosed, the invention is applicable to other ICS signal equipment, railway control circuitry, railway signaling, and railway logic devices, such as, for example, a Microlok II Wayside Control System marketed by Union Switch & Signal, Inc. of Pittsburgh, Pennsylvania.

1.2 OVERVIEW

There are three ways a vital Microlok can communicate: serial communication to the Non-Vital Microlok, serial communication between the Normal/Standby pair, and Vital Input and Output Boards. Since the communication to the Non-Vital Microlok is by definition "non-vital", a breakdown in this communication will have no safety concerns. The serial communication between the Normal/Standby pair utilizes a HEALTH bit that constantly monitors the serial link. The Vital Input and Output Boards are tied together. The inputs to the Vital Input Boards are simply paralleled and the outputs of the Vital Output Boards are "ORed" together with diodes (if both units are permitted to output at all times), or through some type of "Vital OR Gate" (if the customer requires the Standby to suppress its outputs). This means that barring a broken wire (which would result in a failsafe condition) both units will receive the same inputs and, since they have the same logic equations, produce the same outputs. Additional information on the outputs is provided in section 1.3.1.2 Outputs.

The application software addresses the hot standby issue with the above in mind. There is no attempt to actively synchronize all bits at all times, and there is no suppression of outputs from either unit until a failure or disagreement is detected (though it is possible to suppress the Standby outputs if desired by the customer). Since the only purpose of the Standby unit is to provide a hardware backup for the Normal, the Normal unit is considered "boss". If there is a disagreement between the two units the Normal unit will always reset the Standby unit, or the Standby will reset itself, but the Standby can never reset the Normal. This is necessary since there is no way of determining which VITAL Microlok unit is correct, we only know that they are not in agreement. The only way the Normal unit will shut down leaving the Standby unit in control is if the Normal

unit senses an internal failure and takes itself offline. Also, if either unit is reset, all of its outputs are suppressed until they are verified to be in synchronization with the unit currently online.

1.3 IMPLEMENTATION

Since each unit is wired and programmed virtually identical as it would be if it were a stand-alone unit this system is very easy to implement. The only modifications needed to produce the hot standby feature are slight modifications in the hardware and application program.

1.3.1 HARDWARE

1.3.1.1 Inputs

Assuming both Microloks are housed on the same rack, all inputs are single runs to the rack (most likely to a weidmuller). The inputs are then fed in parallel to each unit.

1.3.1.2 Outputs

With both units actively producing outputs at all times, the outputs need only be "ORed" together through a pair, or a series, of diodes (the series of diodes may be needed to meet reliability specifications - see figure 1).

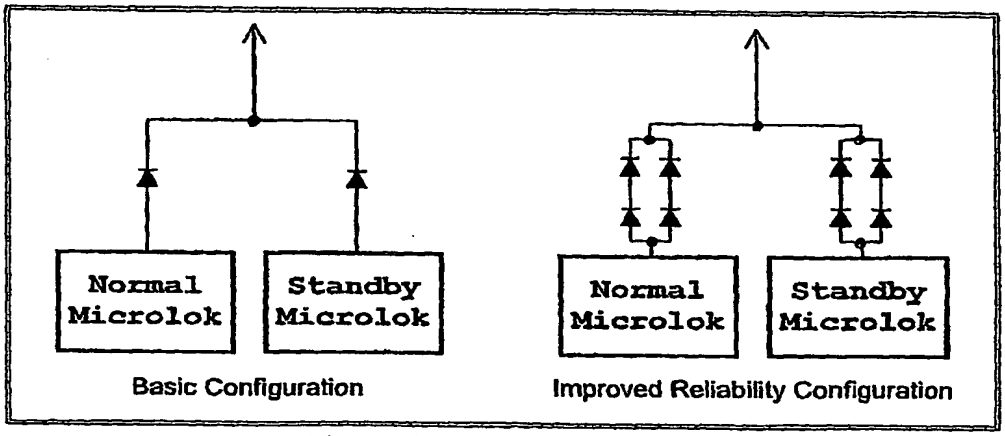

Figure 1: Diode Configurations

However, diodes cannot be used if the customer requires that the outputs of the Standby unit be suppressed if the Normal unit is online. The reason for this is that although Microlok can reliably detect a shorted diode when both units are active, it cannot reliably detect a shorted diode on the Normal unit if the Standby unit is not producing outputs (see Scenarios 8 and 9 in section 1.4.3.3 Diode Faults Induced). If the Standby's outputs are suppressed the outputs must be "ORed" through some type of "Vital OR Gate" which will not allow the possibility of shorting. Union Switch &

Signal's Isolation Module can serve this purpose, but it is not cost effective. A scaled down version could be developed which would provide the necessary protection and be smaller and less expensive.

1.3.1.3 Serial Communication

Two vital serial communications ports are utilized on each unit. This still leaves either two non-vital or one vital and one non-vital port (COM3 can be set up as either vital or non-vital) for links to other units. It is possible to implement the system using only one communication port of each unit but that would require a variation in the application program in each unit. This can easily be accomplished but it is unnecessary unless more than two communication ports are required for links to other units.

1.3.1.4 VCOR Verification

In order to constantly monitor the condition of the other unit, each unit must have a front contact of the other unit's VCOR relay connected to an input of one of its Vital Input Board.

1.3.1.5 Normal Unit Bit

Since the same application program is uploaded into both units, each unit must have one input that is used for identification. This input (on a Vital Input Board) must be constantly high in the Normal unit and constantly low in the Standby unit. The application software uses these constant bit states (Normal - high, Standby - low) in portions of the assign statements that require different operating characteristics for the Normal unit than it does for the Standby unit.

1.3.2 APPLICATION PROGRAM

Any application program designed for a stand alone unit can be changed to a hot standby application simply by adding three logic systems and modifying all output bits to be one of three types. Also, if external Lock relays are not utilized, the internal Lock bits must be modified as if they were outputs.

Logic Systems

The three logic systems: Synchronization, Health, and Reset serve to restrict, maintain, and protect the operation of the hot standby system. More detailed explanations of the bits that comprise these systems can be found in the test program in the Appendix.

Synchronization

The Synchronization system restricts the unit from producing outputs if the other unit is already online and the output states of the units disagree. Once the unit achieves synchronization it is permitted to produce outputs and the Synchronization system is not utilized until the unit is reset and attempts to come back on line. This system is equally functional in both the Normal and Standby units.

The Synchronization system employs the following bits:

- STAND.ALONE.SYNC.DELAY is a slow set bit that provides a 1 second delay for the unit to stabilize before the other unit's VCOR is referenced.
- STAND.ALONE.SYNC sets the SYNC bit if STAND.ALONE.SYNC.DELAY is set and the other unit's VCOR is down.
- SYNC.WAIT is a slow set bit which forces the unit coming online to wait until serial communication is stabilized before attempting to synchronize.
- SYNC is the controlling bit. When the unit coming online is synchronized with the unit currently online the SYNC bit is set.

Health

The Health system is verified by the constant exchange of a HEALTH bit over the serial communication line. When the Normal unit's VCOR is picked, the HEALTH bit is required for the Standby unit to stay online. Without the HEALTH bit verifying that serial communication is stable, the Standby unit is reset. This insures that if communication is lost, one unit is taken offline (the Standby). Though this system is primarily utilized in the Standby portions of the assign statements, the Normal unit also uses the HEALTH.WAIT bit to maintain its Restricted bits (see below) while the Standby unit is coming online.

The HEALTH system employs the following bits:

- HEALTH.WAIT.DELAY is a slow clear bit that is set when the other unit's VCOR is picked but serial communication is not yet established. Its function is to set HEALTH.WAIT and maintain it until either SL.IN.HEALTH is received or time expires.
- HEALTH.WAIT is a slow set bit that sets 1 second after HEALTH.WAIT.DELAY. It clears when SL.IN.HEALTH is received or time expires and HEALTH.WAIT.DELAY clears.
- SL.OUT.HEALTH is the serial bit that the unit sends the other unit.
- SL.IN.HEALTH is the serial bit that the unit receives in from the other unit.

Reset

The Reset system protects the pair's vital functions by forcing the Standby unit to reset when there is a disagreement between the units. This system is always active in the Standby unit if the Normal unit's VCOR is picked.

The RESET system employs the following bits:

- SYS.RESET is a slow set bit that is only operational in the Standby unit. When the bit sets, the unit resets.
- SL.OUT.RESET is sent from the Normal unit to the Standby unit when the Normal determines there is a disagreement and wants the Standby to reset.
- SL.IN.RESET is the bit the Standby unit receives when the Normal unit sends SL.OUT.RESET.
- GROUP.XX.RESET type bits are groups of individual reset bits that are used to simplify the SYS.RESET assign statement and eliminates the need for timers on all individual reset bits.

Bit Types

There are three types of bits: Unrestricted, Half Restricted, and Restricted. All three of these types must be utilized to insure that the hot standby operates safely but does not waste system resources on unnecessary tasks.

These three types of bits have the following in common:

- If the other unit's VCOR is down the unit will produce the output whenever the assign statement is satisfied.
- If the other unit's VCOR is up the unit must also receive serial communication.
    - Unrestricted bits require a generic health bit.
    - Half-Restricted bits require a bit verification from Normal to Standby.
    - Restricted bits require bit verification to and from both units.
- If the other unit is in control, the unit being brought online cannot produce any outputs until it is in SYNC.
- If both units are online and any bit states disagree for a selected period of time either the Normal unit will reset the Standby or the Standby will reset itself.
- With minor modifications the Standby unit's outputs can be suppressed when the Normal unit's VCOR is picked.

Unrestricted

These bits require no bit specific serial communication between the units in order to produce an output; therefore they are the fastest and should always be utilized whenever possible. They should never be used for signal lighting or Locks.

Half Restricted

These bits are unrestricted in the Normal unit, but restricted in the Standby. The Standby unit cannot produce the output until it receives verification (via serial communication) that the Normal unit has also satisfied the assign statement. This type of bit is specifically designed for signal lighting. If the bits are out of sync, it can only be that the Normal unit has the aspect lit and the Standby does not. In this event the Standby unit is reset, and the signal aspect does not change.

Restricted

These bits are restricted in both the Normal and the Standby units. Neither unit can produce the output until it receives verification (via serial communication) that the other unit has also satisfied the assign statement. This type of bit is the slowest due to the amount of serial communication involved. It is specifically designed for Locks. The bit cannot be set (unlocked) until both units satisfy the assign statement and it will be cleared (locked) immediately at any time the units do not agree. There are two considerations concerning this type of bit:

1. If locking is performed without the use of external Lock relays the internal variables will require this configuration.

2. If the response time is too long due to the use of serial communication the verification will need to be passed via Vital Input and Output Boards.

1.4 TEST PLAN

1.4.1 BACKGROUND

Both Microlok units were housed in a cabinet and shared the same Power Supply. The serial communication between the units was accomplished with a cable from the Normal CPU to the Standby CPU. The communication cable tied Normal COM1 to Standby COM2, and Normal COM2 to Standby COM1. Inputs were paralleled to both units but the wires going to the Standby unit were clipped together and could be disconnected. The outputs were "ORed" together with diodes, as shown in the Basic Configuration example in Figure 1.

1.4.2 BASELINE INFORMATION

1.4.2.1 Software

Both units were uploaded with the following application program:

File Name:	MLK_TEST_2.0.doc
Application Image CRC:	e6c7
Application Image Checksum:	f78b

1.4.2.2 Reference Points

The following references were used for testing purposes:

- "VCOR picked" was referenced from the lighting of the VCOR indication on the Power Supply Board.
- Serial communication was referenced from the COM indications (A, B, C, D, and E) on the CPU Board.
- Outputs were referenced from the indications on the Vital Output Board.
- For the purpose of testing, the following reference bits were not considered to be outputs:
  - OUT 7 - SYNC.WAIT
  - OUT 8 - SYNC
  - OUT 9 - HEALTH.WAIT.DELAY
  - OUT 10 - HEALTH.WAIT
  - OUT 12 - SL.OUT.04
  - OUT 13 - SL.IN.04
  - OUT 14 - OUT.RESET
  - OUT 15 - IN.RESET
  - OUT 16 - COMALT

1.4.3 TESTS

1.4.3.1 No Faults Induced

Scenario 1	Unit RESET while the other is offline (CPU pulled)
Purpose	To verify that each unit can operate as a stand-alone unit.
Attempts	5 (each unit)
Result	Unit came online and produced outputs: • Time from RESET to picking of VCOR = 14 sec. • Time from RESET to outputs = 15 sec. • Time from RESET to attempted serial communication = 25 sec.
Comment	• The times were the same for both the Normal and Standby units.

Scenario 2	Unit RESET while the other unit is online, communicating, and producing outputs
Purpose	To verify that each unit can be brought online without any interruption of controlling unit.
Attempts	5 (each unit)
Result	Unit came online and produced outputs: • Time from RESET to picking of VCOR = 14 sec. • Time from RESET to serial communication = 25 sec. • Time from RESET to outputs = 30 sec.
Comment	• The times were the same for both the Normal and Standby units. • No change in outputs occurred in the other unit.

Scenario 3	Power up both units simultaneously
Purpose	To verify that there is no circular logic which would prevent the units from coming online simultaneously.
Attempts	5
Result	Both units powered up and produced outputs: • Time from power to picking of VCOR = 18 sec. • Time from power to serial communication = 29 sec. • Time from power to both unit's outputs = 35 sec.
Comment	• This functioned as expected. • No interruption of outputs occurred in either unit.

1.4.3.2 Faults Induced

Scenario 4	• Both units online • Remove IN.02 bit from the Standby unit.
Purpose	To prove that the Normal unit will RESET the Standby unit if there is a disagreement in bit states, and that there is no danger in allowing the Standby unit to attempt to come back online after it is RESET.
Attempts	• 1 disconnect of input • Standby unit RESET 34 times • 20 minutes time
Result	Standby unit RESET and continued to cycle: • Time from removal of input to first RESET = 10 sec. • Time from removal of input to first VCOR pick = 23 sec. • Time from removal of input to first establishment of serial communication with Normal unit = 35 sec. • Time from removal of input to second RESET = 38 sec.
Comment	• Standby unit continued to cycle, resetting every 15 seconds after the VCOR picked. • The Standby unit never produced any outputs • The Normal unit's outputs were never interrupted. • After approximately 30 minutes IN.02 was restored to Standby unit and after the completion of its current reset, it came back online and produced outputs.

Scenario 5	• Both units online • Remove IN.02 bit from the Standby unit • Reset Normal unit • Wait 1 minute 30 seconds • Restore IN.02 bit to the Standby unit
Purpose	To prove that synchronization is required for the Normal to come online if the Standby is in control and that the Normal unit cannot RESET the Standby as the Normal unit comes online.
Attempts	5
Result	Normal unit did not produce outputs until after IN.02 was restored to the Standby unit: • Time from RESET to VCOR pick = 14 sec. • Time from RESET to serial communication = 25 sec. • Time from RESET to IN.02 restored to Standby = 1 min. 30 sec. • Time from RESET to outputs = 1 min. 30 sec.
Comment	• The Standby unit's outputs were never interrupted.

Scenario 6	• Both units online • RESET unit while IN.02 and IN.04 are constantly toggled (approximately 4 times a second)
Purpose	To prove that flashing bits or several bits changing state will not hinder the unit from synchronizing and coming online.
Attempts	5 (each unit)
Result	• Both units came online in the usual timeframe.
Comment	• The controlling unit's outputs were never interrupted.

Scenario 7	• Both units online • Bits OUT.02 and OUT.04 set, and bit OUT.03 clear • RESET unit • Wait 5 seconds • RESET other unit
Purpose	To prove that either unit will produce outputs immediately if the other unit's VCOR is down.
Attempts	5 (each unit)
Result	• The unit RESET first came online and immediately produced outputs. • The unit RESET second had to synchronize before it produced outputs.
Comment	• The controlling unit's (the unit RESET first) outputs were never interrupted as the second unit came online.

1.4.3.3 Diode Faults Induced

Scenario 8	• Both units online • Place a short across one output diode
Purpose	To prove that a shorted diode can be detected by Microlok
Attempts	• Twice on each diode of OUT.02, OUT.03 and OUT.04 with the outputs high. • Twice on each diode of OUT.02, OUT.03 and OUT.04 with the outputs low. • A total of 24 tests.
Result	• When the outputs were high, both units detected the short in 9-10 seconds and RESET. • When the outputs were low the Normal unit detected the short within 1 second and RESET. • When the outputs were low the Standby unit did not detect the short. All outputs were shorted for at least 1 minute and one was allowed to remain shorted for over 8 minutes and the unit still failed to detect it. After the elapsed time, the output was toggled high and the short was detected within 10 seconds.
Comment	• The conclusion is that Microlok can consistently detect a shorted diode within 10 seconds if both outputs are high. • The unit without the shorted diode maintained its output without interruption. • Since some properties (possibly MARTA included) would require the Standby unit's outputs to be suppressed if the Normal unit's VCOR is picked, the question still remains if Microlok can detect a shorted diode if the outputs are not the same (both high or both low). See Scenario 9.

Scenario 9	• Install modified program in both Microloks which suppresses the Standby's outputs if the Normal's VCOR is picked. • Bring both units online (the Normal unit has outputs the Standby does not) • Place a short across one output diode
Purpose	To determine if shorted diode can be detected by Microlok if the outputs are not the same (both high or both low)
Attempts	• Twice on each diode of OUT.02, OUT.03 and OUT.04 • A total of 12 tests.
Result	• When a diode of the Standby unit was shorted the Standby unit detected the short within 1 second and RESET. • When a diode of the Normal unit was shorted the Normal unit did not detect the short. All outputs were shorted for at least 1 minute and one was allowed to remain shorted for over 8 minutes and the unit still failed to detect it. After the elapsed time, the Normal unit was RESET and when it attempted to come back online (its outputs low and the Standby unit's outputs high) it detected the short and continually RESET.
Comment	• The conclusion is that Microlok cannot detect a shorted diode when it is on the high output and the other output is low. • If the Standby unit's outputs must be suppressed when the Normal units VCOR is picked then diodes are not a viable option.

1.4.4 CONFIGURATION OF TEST UNITS

NORMAL UNIT				
BOARD	SLOT	PART #	SERIAL #	REVISION
CPU	18	N17003401	1201028	2
Vital Input - 16	13	N17001001	0998048	2
Vital Output - 16	15	N17000501	0898006	2
Power Supply	6	N16600301	1998022	4

STANDBY UNIT				
BOARD	SLOT	PART #	SERIAL #	REVISION
CPU	18	N17001301	3100010	7
Vital Input - 16	13	N17001001	0998033	2
Vital Output - 16	15	N17000501	0898005	2
Power Supply	10	N451810750	2498001	2

Appendix 1 - Sample Program

```
/*****************************************************************************************
**                                                                                     **
**                                                                                     **
** PROGRAM:    HOT STANDBY TEST                                                        **
** LOCATION:   QA LAB                                                                  **
** UNIT:       MICROLOK II                                                             **
** AUTHOR:     JOSEPH S. BLEVINS                                                       **
** FILE NAME:  MLK_TEST_2.0.doc                                                        **
** REV:        2.0                                                                     **
**                                                                                     **
** REV HISTORY:        1 -     Modified SYNC system.                                   **
**                         -   Created STAND.ALONE.SYNC.DELAY, and                     **
**                             STAND.ALONE.SYNC.                                       **
**                         -   Added STAND.ALONE.SYNC to the SYNC                      **
**                             assign statement so it will set 1 second                **
**                             after the unit comes online if the other                **
**                             unit's VCOR is down.                                    **
**                         -   Removed "+ (SL.OUT.XX * ~VCOR) from all                 **
**                             OUT.XX.SYNC assign statements.                          **
**                                                                                     **
**                                                                                     **
**                                                                                     **
**                                                                                     **
** VER:         0.00                                                                   **
** APPLICATION IMAGE CRC:             e6c7                                             **
** APPLICATION IMAGE CHECKSUM:   f78b                                                  **
**                                                                                     **
** CHECKED BY:                                                                         **
** TESTED BY:                                                                          **
** INSERVICE BY:                                                                       **
**                                                                                     **
*****************************************************************************************/

MICROLOK_II PROGRAM HOT_STANDBY_MLK_TEST_2.0;
INTERFACE

/*********************************************************************/
```

LOCAL

/**** VITAL OUTPUT BOARD 1 ****/

BOARD: VO_SLOT_J15
ADJUSTABLE ENABLE: 1
TYPE: OUT16
OUTPUT:

> Only OUT.02, OUT.H.03, and OUT.L.04 simulate true outputs. The remaining bits are only used to give an indication on the output board for testing purposes.

SPARE,	OUT.02,	OUT.H.03,	OUT.L.04,
SPARE,	SPARE,	OUT.SYNC.WAIT.07,	OUT.SYNC.08,
OUT.HEALTH.WAIT.DELAY.09,	OUT.HEALTH.WAIT.10,	SPARE,	OUT.OUT.L.04.12,
OUT.IN.L.04.13,	OUT.OUT.RESET.14,	OUT.IN.RESET.15,	OUT.COMALT.16;

/**********/

/* VITAL INPUT BOARD 1 ****/***

BOARD: VI_SLOT_J13
ADJUSTABLE ENABLE: 1
TYPE: IN16
INPUT:

> In the Normal unit, NORMAL (bit 16) is energized from a constant source. It must be high in the Normal unit and low in the Standby unit.

VCOR,	IN.02,	IN.03,	IN.04,
SPARE,	SPARE,	SPARE,	SPARE,
SPARE,	SPARE,	SPARE,	SPARE,
SPARE,	SPARE,	SPARE,	NORMAL;

/**********/

COMM

Two COM ports are used so that the same software can be used in both units.

/**** BITS IN/OUT MASTER ****/

LINK: HOT_MASTER
ADJUSTABLE ENABLE: 1
PROTOCOL: MICROLOK.MASTER
ADJUSTABLE POINT.POINT: 1;
ADJUSTABLE PORT: 1;
ADJUSTABLE BAUD: 19200;
ADJUSTABLE STOPBITS: 1;
ADJUSTABLE PARITY: NONE;
ADJUSTABLE KEY.ON.DELAY: 0;
ADJUSTABLE KEY.OFF.DELAY: 12;
ADJUSTABLE STALE.DATA.TIMEOUT: 3:SEC;
ADJUSTABLE POLLING.INTERVAL: 50:MSEC;
ADJUSTABLE MASTER.TIMEOUT: 100:MSEC;
ADDRESS: 1
ADJUSTABLE ENABLE: 1

/**** BITS OUT ****/

OUTPUT:

SL.OUT.02, SL.OUT.03, and SL.OUT.04 represent all output bits. If the Microlok had two 16 bit Vital Ouput Boards (and all bits were used) then there would be 32 bits listed. SL.OUT.HEALTH and SL.OUT.RESET are the only extra bits required for hot standby operation.

SL.OUT.02, SL.OUT.H.03, SL.OUT.L.04,

SL.OUT.HEALTH, SL.OUT.RESET;

/**** BITS IN/OUT SLAVE ****/

LINK: HOT_SLAVE
ADJUSTABLE ENABLE: 1
PROTOCOL: MICROLOK.SLAVE
ADJUSTABLE POINT.POINT: 0;
ADJUSTABLE PORT: 2;
ADJUSTABLE BAUD: 19200;
ADJUSTABLE STOPBITS: 1;
ADJUSTABLE PARITY: NONE;
ADJUSTABLE KEY.ON.DELAY: 0;
ADJUSTABLE KEY.OFF.DELAY: 12;
ADJUSTABLE STALE.DATA.TIMEOUT: 3:SEC;
ADDRESS: 1
ADJUSTABLE ENABLE: 1

/**** BITS IN ****/

INPUT:

> SL.IN.02, SL.IN.03, and SL.IN.04 represent all output bits from the other unit. If the Microlok had two 16 bit Vital Ouput Boards (and all bits were used) then there would be 32 bits listed.
> SL.OUT.HEALTH and SL.OUT.RESET are the only extra bits required for hot standby operation.

SL.IN.02, SL.IN.H.03, SL.IN.L.04,

SL.IN.HEALTH, SL.IN.RESET;

/**********/

/**** VARIABLES ****/

BOOLEAN BITS

SYS.RESET, OUT.RESET.02,	GROUP.01.RESET, OUT.H.RESET.03,	GROUP.02.RESET, OUT.L.RESET.04,	GROUP.03.V.RESET,
SYNC, OUT.02.SYNC,	SYNC.WAIT, OUT.H.03.SYNC,	STAND.ALONE.SYNC.DELAY, OUT.L.04.SYNC,	STAND.ALONE.SYNC,
HEALTH.WAIT.DELAY,	HEALTH.WAIT,		
COMALT,			
SL.IN.H.03.D,	SL.IN.L.04.D;		

/**********/

TIMER BITS

/**** TIMERS ****/

SL.OUT.RESET is sent from the Normal unit to the Standby unit when the Normal determines there is a disagreement in bit states. It is delayed to allow the Standby time to synchronize. The exact setting for this bit is based on the needs of each application. It should be as short as possible without effecting reliability.

SL.OUT.RESET:	SET = 3:SEC	CLEAR = 0:SEC;

SYS.RESET is an internal bit that RESETS the Standby unit if it is out of synchronization with the online Normal unit. It is slightly delayed to insure the unit does not falsely reset. The exact setting for this bit is based on the needs of each application. It should be as short as possible without effecting reliability.

SYS.RESET:	SET = 3:SEC	CLEAR = 0:SEC;

The GROUP.RESET bits represent groups of individual bit reset commands. They are slightly delayed to insure the unit does not reset falsely. GROUP.03.V.RESET contains bits that are "more vital" such as Switch Locking or Route Locking.

therefore they are given a shorter reset time. The exact setting for these bits is based on the needs of each application. They should be as short as possible without effecting reliability.

GROUP.01.RESET:	SET = 3:SEC	CLEAR = 0:SEC;
GROUP.02.RESET:	SET = 3:SEC	CLEAR = 0:SEC;
GROUP.03.V.RESET:	SET = 1:SEC	CLEAR = 0:SEC;

HEALTH.WAIT.DELAY is an internal bit that works together with HEALTH.WAIT to allow the Standby unit to maintain its outputs while the Normal is brought online. It is also utilized in the Normal unit but only for Restricted bits. It fills in the gap between the time the Normal's VCOR picks and communication between the pair is established. It is set for 20 seconds because it takes approximately 15 seconds for a unit to establish serial communication after the VCOR is picked.

HEALTH.WAIT.DELAY:	SET = 0:SEC	CLEAR = 20:SEC;

HEALTH.WAIT shortens the effect of HEALTH.WAIT.DELAY to 1 second after serial communication is established. HEALTH.WAIT is used in all output bit assign statements.

HEALTH.WAIT:	SET = 0:SEC	CLEAR = 1:SEC;

STAND.ALONE.SYNC.DELAY is a slow set bit that allows the unit to stabilize before VCOR is referenced for SYNC.

STAND.ALONE.SYNC.DELAY:	SET = 1:SEC	CLEAR = 0:SEC;

SYNC.WAIT is a slow set internal bit that allows serial communication to stabilize after the unit is powered up before synchronization is verified. It should always be set for 5 seconds or longer.

SYNC.WAIT:	SET = 5:SEC	CLEAR = 0:SEC;

SL.IN.H.03.D and SL.IN.L.04.D are test bits used to simulate the delay in serial communications between the units.

SL.IN.H.03.D:	SET = 1:SEC	CLEAR = 0:SEC;
SL.IN.L.04.D:	SET = 1:SEC	CLEAR = 0:SEC;

```
/*************************************************************/

CONSTANTS BOOLEAN

        ONE    = 1;
        ZERO   = 0;

/*************************************************************/

CONFIGURATION
        SYSTEM

        ADJUSTABLE DEBUG_PORT_ADDRESS:      1;
        ADJUSTABLE DEBUG_PORT_BAUDRATE:     9600;
        ADJUSTABLE LOGIC_TIMEOUT:           2:SEC;
        ADJUSTABLE DELAY_RESET:             3:SEC;

/*************************************************************/
```

/**** LOGIC ****/

LOGIC BEGIN

/**** MICROLOK SYSTEM CIRCUIT ****/

ASSIGN	ONE	TO	CPS.ENABLE;
ASSIGN	ONE	TO	STAND.ALONE.SYNC.DELAY;
ASSIGN	ONE	TO	SL.OUT.HEALTH;
ASSIGN	SYS.RESET	TO	RESET;

/*** RESET BITS ***/

```
"  SYS.RESET is a slow set bit that only functions in the Standby unit.
   The bits in the assign statement function as follows:
   VCOR insures that the unit will only RESET if the Normal unit is online.
   SL.IN.RESET comes from the Normal unit and forces the Standby unit to RESET.
   ~SL.IN.HEALTH insures the Standby unit will RESET itself if serial communication
   is lost between the units.
   ~HEALTH.WAIT insures the Standby unit will not RESET itself before serial
   communication is established when the Normal unit is coming online.
   GROUP.01.RESET, GROUP.02.RESET, and GROUP.03.V.RESET are groups of
   individual reset bits. See below.                                          "
```

```
ASSIGN  (~NORMAL * VCOR) *
        (SL.IN.RESET +
        ((~SL.IN.HEALTH * ~HEALTH.WAIT) +                    TO  SYS.RESET;
        (GROUP.01.RESET + GROUP.02.RESET +
        GROUP.03.V.RESET)))
```

```
"  SL.OUT.RESET is a slow set bit that is sent from the Normal unit to the Standby
   when any output bit is out of sync. It is primarily controlled by the GROUP.RESET
   bits, however, the SYNC bit is also required so that the Normal unit cannot reset
   the Standby unit if the Normal is being powered up and cannot achieve
   synchronization with the Standby.                                          "
```

```
ASSIGN  (NORMAL * SYNC) *
        (GROUP.01.RESET + GROUP.02.RESET + GROUP.03.V.RESET)   TO  SL.OUT.RESET;
```

GROUP.01.RESET, GROUP.02.RESET, and GROUP.03.V.RESET are groups of individual reset bits (though only one RESET bit is assigned to each for testing). The individual reset bits are grouped together to simplify the SYS.RESET equation and to allow for a longer time delay for non-synchronous situations which may be caused by serial communication delays. Three groups are used for testing but the maximum number of groups is unlimited. Multiple groups should be used to limit the number of bits so that continuous changes of bit states will not by misinterpreted as a non-synchronous condition. GROUP.03.V.RESET represents groups of bits that are "more vital" such as Switch Locks and Route Locks. This group is given the absolutely shortest time delay possible to maintain reliability.

ASSIGN	OUT.RESET.02	TO	GROUP.01.RESET;
ASSIGN	OUT.H.RESET.03	TO	GROUP.02.RESET;
ASSIGN	OUT.L.RESET.04	TO	GROUP.03.V.RESET;

/*** SYNCHRONIZATION BITS ***/

SYNC suppresses all outputs of the unit being brought online until they are verified to be synchronous with the unit currently in control or the other unit's VCOR is down. Once it is set it is stuck high until the unit is powered down.

ASSIGN	SYNC + STAND.ALONE.SYNC + (OUT.02.SYNC * OUT.H.03.SYNC * OUT.L.04.SYNC)	TO	SYNC;

SYNC.WAIT is a slow set bit that suppresses verification of bits in the unit being brought online until it is powered up and both the unit and the serial communication link are stable. Once it is set it is stuck high until the unit is powered down.

ASSIGN	SYNC.WAIT + (VCOR * SL.IN.HEALTH)	TO	SYNC.WAIT;

STAND.ALONE.SYNC is an internal bit that will set the SYNC bit one second after the unit is powered up if the other unit's VCOR is down.

ASSIGN	~VCOR * STAND.ALONE.SYNC.DELAY	TO	STAND.ALONE.SYNC;

/*** HEALTH BITS ***/

HEALTH.WAIT and HEALTH.WAIT.DELAY allow the Standby unit to maintain its outputs between the time that the Normal VCOR picks and communication is established between the units. HEALTH.WAIT.DELAY is a slow clear bit that sets when the VCOR picks in the unit coming online. HEALTH.WAIT.DELAY sets HEALTH.WAIT which remains high until either HEALTH.WAIT.DELAY expires or serial communication is established. This is necessary to insure that as soon as serial communication is established the HEALTH bit is not able to override any logic.

ASSIGN	VCOR * ~SL.IN.HEALTH * ~HEALTH.WAIT.DELAY	TO	HEALTH.WAIT.DELAY;
ASSIGN	HEALTH.WAIT.DELAY * ~SL.IN.HEALTH	TO	HEALTH.WAIT;

/*** OUTPUT BITS ***/

/* UNRESTRICTED BIT ***/**

The term IN.02 is used for testing purposes. In reality it would be replaced by a logic equation. In the term SL.OUT.02, SL stands for Serial Link, and OUT.02 represents the resulting bit of the satisfied assign statement. SL.OUT.02 is immediately sent out serially to the other unit.

ASSIGN	IN.02	TO	SL.OUT.02;

OUT.02.SYNC is primarily satisfied by the SL.OUT.02 bit. If the other unit is online (VCOR is picked, or in the process of booting up) it is referenced to insure that the bit is in the same state. Once it is set it is stuck high until the unit is powered down. If the other unit is offline (VCOR down) this bit is bypassed and the SYNC bit is satisfied with STAND.ALONE.SYNC.

ASSIGN	OUT.02.SYNC + (((SL.OUT.02 * SL.IN.02) + (~SL.OUT.02 * ~SL.IN.02)) * VCOR * SYNC.WAIT)	TO	OUT.02.SYNC;

OUT.02 is the bit that sets the output high on the Vital Output Board. It is primarily satisfied by the SL.OUT.02 bit. In the Normal unit the only other requirement is that the SYNC bit must be set (which it will be unless the Normal is in the process of coming online). The Standby unit requires a serial communication HEALTH bit or HEALTH.WAIT. HEALTH.WAIT is used keep the Standby unit's outputs set between the time the Normal unit's VCOR picks and serial communication is established when the Normal unit is being brought online. Both the Normal and Standby units will immediately set the bit if SL.OUT.02 is high and the other unit is offline (VCOR down).

ASSIGN	SL.OUT.02 * (SYNC * (NORMAL + (~NORMAL * (SL.IN.HEALTH + HEALTH.WAIT))) + ~VCOR)	TO	OUT.02;

OUT.RESET.02 can only be set high in the Normal unit. Its purpose is to cause the Standby unit to reset if there is a disagreement in the bit state between the units.

ASSIGN	NORMAL * ((SL.OUT.02 * ~SL.IN.02) + (~SL.OUT.02 * SL.IN.02))	TO	OUT.RESET.02;

/* HALF RESTRICTED BIT ***/**

The logic statement for OUT.H.03 functions the same as OUT.02 above, with one exception. In the statement for OUT.H.03 the generic serial communication HEALTH bit is replaced with the corresponding bit (SL.IN.H.03.D) from the Normal unit. This suppresses the output from the Standby unit until it has been verified that the Normal has also satisfied the assign statement. SL.IN.H.03.D is a slow set bit used for testing to simulate serial communication delays.

ASSIGN	IN.03	TO	SL.OUT.H.03;
ASSIGN	OUT.H.03.SYNC + (((SL.OUT.H.03 * SL.IN.H.03) + (~SL.OUT.H.03 * ~SL.IN.H.03)) * VCOR * SYNC.WAIT)	TO	OUT.H.03.SYNC;
ASSIGN	SL.OUT.H.03 * (SYNC * (NORMAL + (~NORMAL * (SL.IN.H.03.D + HEALTH.WAIT))) + ~VCOR)	TO	OUT.H.03;
ASSIGN	NORMAL * ((SL.OUT.H.03 * ~SL.IN.H.03) + (~SL.OUT.H.03 * SL.IN.H.03))	TO	OUT.H.RESET.03;

/*** RESTRICTED BIT ***/

The logic statements for OUT.L.04 are the same as OUT.H.03 above, with one exception. In the statement for OUT.L.04 there are no separate variables for Normal or Standby. Both units must have SL.OUT.L.04, be in SYNC, and receive the corresponding bit from the other unit. This suppresses the output from either unit until it has been verified that the other unit has also satisfied the assign statement and immediately drops the output if it loses the verification from the other unit.

ASSIGN	IN.04	TO	SL.OUT.L.04;
ASSIGN	OUT.L.04.SYNC + (((SL.OUT.L.04 * SL.IN.L.04) + (~SL.OUT.L.04 * ~SL.IN.L.04)) * VCOR * SYNC.WAIT)	TO	OUT.L.04.SYNC;
ASSIGN	SL.OUT.L.04 * (SYNC * (SL.IN.L.04.D + HEALTH.WAIT) + ~VCOR)	TO	OUT.L.04;
ASSIGN	NORMAL * ((SL.OUT.L.04 * ~SL.IN.L.04) + (~SL.OUT.L.04 * SL.IN.L.04))	TO	OUT.L.RESET.04;

/*** COMMUNICATION ALERT ***/

COMALT may be used to alert central control of any problems in the communication between the Normal and Standby units.

ASSIGN	~SL.IN.HEALTH	TO	COMALT;

/*** FOR TEST PURPOSES ONLY ***/

/*** COM DELAY SIMULATORS ***/

SL.IN.H.03.D and SL.IN.L.04.D are slow set bits that simulate the possible delay in the serial communication between the Normal and Standby units.

ASSIGN	SL.IN.H.03	TO	SL.IN.H.03.D;
ASSIGN	SL.IN.L.04	TO	SL.IN.L.04.D;

/*** BIT MONITORS ***/

The following bits produce indications on the Vital Output Board for testing purposes.

ASSIGN	SYNC.WAIT	TO	OUT.SYNC.WAIT.07;
ASSIGN	SYNC	TO	OUT.SYNC.08;
ASSIGN	HEALTH.WAIT.DELAY	TO	OUT.HEALTH.WAIT.DELAY.09;
ASSIGN	HEALTH.WAIT	TO	OUT.HEALTH.WAIT.10;
ASSIGN	SL.OUT.L.04	TO	OUT.OUT.L.04.12;
ASSIGN	SL.IN.L.04	TO	OUT.IN.L.04.13;
ASSIGN	SL.OUT.RESET	TO	OUT.OUT.RESET.14;
ASSIGN	SL.IN.RESET	TO	OUT.IN.RESET.15;
ASSIGN	COMALT	TO	OUT.COMALT.16;

/**********/

END LOGIC
END PROGRAM

Minimum Hardware Requirements for Hot Standby

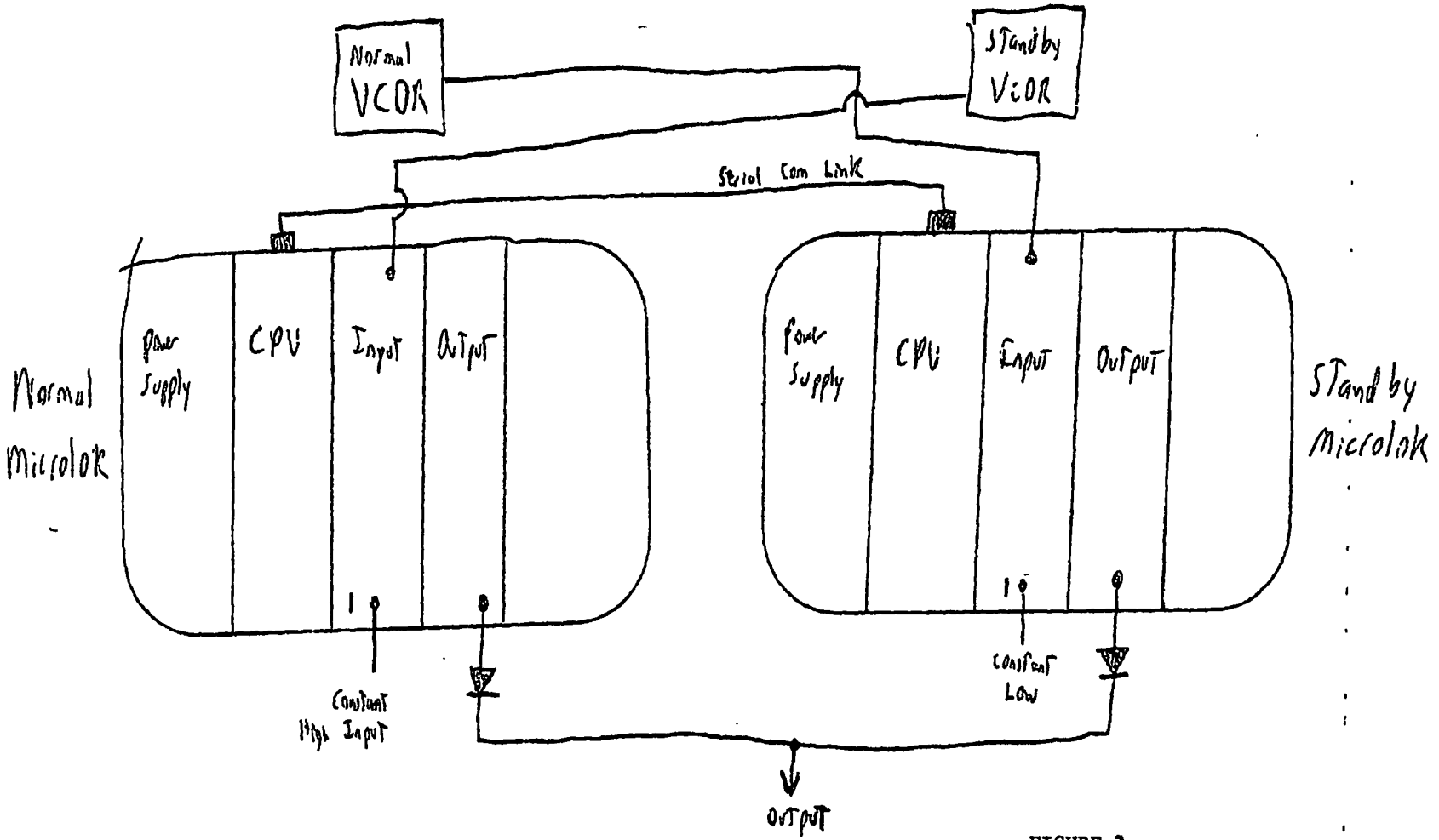

FIGURE 2.